a/Brief 5/21/04
1601
1003

Hewlett Packard Docket No.: 10011974-1

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
Board of Patent Appeals and Interferences

| | |
|---|---|
| In re Application of: **Calvin Selig, et al.** | Art Unit: **2676** |
| Serial No.: **09/823,660** | Examiner: **Tran, Tam D.** |
| Filed: **March 31, 2001** | |
| For: **Fast Clear Technique for Regions Having Subregions** | |



APPEAL BRIEF

Assistant Commissioner for Patents

Washington, D.C. 20231

Sir:

Applicant (hereinafter "Appellant") appeals from examiner Tran's final rejection mailed 11/17/2003 (hereinafter "the final rejection"). A Notice of Appeal was filed by facsimile on 3/17/2004 with a petition and fee for a one-month extension of time. Appellant believes no further extensions of time are required; but if further extensions are required, the Commissioner is hereby petitioned to grant such extensions and to charge the requisite fees to Appellant's deposit account no. 08-2025.

Real party in interest

The real party in interest in this appeal is the assignee of the subject patent application: Hewlett-Packard Development Company, a Texas limited partnership having its principal place of business in Houston.

Related appeals and interferences

Also under appeal is Appellant's patent application serial number 09/823,505, filed March 31, 2001, titled "Technique for Eliminating Stale Information from a Computer Graphics Buffer" (hereinafter "the '505 application"). The '505 application and the present application are related in the sense that they disclose related subject matter. They are not related as parent/child, divisionals, continuations or continuations-in-part.

Status of claims

Claims 1-50 are pending as they were originally presented. They have been finally rejected and are the subject of this appeal.

Status of amendments

No amendments were submitted before or after the final rejection.

Concise summary of invention

"Fast clear" is a conventional computer graphics technique wherein a value can effectively be set for all of the pixels in a region without actually writing the value to each of the memory locations corresponding to the pixels. Appellant has improved on conventional fast clear techniques: Appellant claims a novel and efficient method for performing clear operations in regions having subregions--for example, in windows having subwindows.

The inventive method performs an initialization routine in response to a first clear command directed to a region. The initialization routine eliminates stale information from all pixels in the region that are outside a subregion. Then, responsive to subsequent clear commands, the method (1) updates a clear count for the region, and (2) causes clear count values for each pixel outside the subregion to equal the updated clear count for the region. The method has the effect of providing fast clear for pixels inside the subregion while optionally preserving display of previous contents for pixels outside the subregion.

Issues

The issue on appeal is whether claims 1-50 are anticipated by U.S. patent 5,805,868 ("Murphy") under 35 U.S.C. § 102(b) .

Grouping of claims

Claims 1, 4-9 and 47 will stand or fall together.

Claims 15, 20-25 and 48 will stand or fall together.

Claims 31, 36-41 and 50 will stand or fall together.

All other claims will be argued separately.

Argument

THE EXAMINER'S FINDING OF ANTICIPATION IS ERRONEOUS BECAUSE AT LEAST ONE ELEMENT IN EACH OF CLAIMS 1-50 IS ABSENT FROM THE MURPHY REFERENCE.

1. The Murphy Reference

In the context of fast clear, the Murphy reference discloses nothing more than the well-known "striping" technique that is described in Applicant's own specification and acknowledged therein as conventional:

> Any conventional fast clear technique, such as a striping technique, may be utilized to handle clear commands for region 800. In accordance with a striping technique, region 800 may be divided into stripes 802. Alternatively, region 800 may be divided into vertical columns, or region 800 may be further divided into a matrix of rectangles. (The terms "subdivisions" and "stripes" and "striping" are used interchangeably herein to refer to any and all of these alternative stripes, columns or rectangles.) Responsive to each clear command, the current clear count for region 800 may be incremented, and then an actual clear may be performed in one of stripes 802. The one stripe 802 chosen for actual clearing may change according to a cyclic schedule so that all of stripes 802 will have been actually cleared at the completion of the cycle. The benefit of the striping technique is that an actual full clear of the entire striped area need not be performed in response to any one clear command, provided the number of stripes in the region is not greater than the maximum value of the current clear count register for the region; instead, the full clear is amortized over several clear commands.

Application at page 12, lines 8-22. Compare the above with Murphy's disclosure:

> The fast clear mechanism provides a method where the cost of clearing the depth and stencil buffers can be amortized over a number of clear operations issued by the application. This works as follows:
>
> The window is divided up into n regions, where n is the range of the frame counter (16 or 256). Every time the application issues a clear command the reference frame counter is incremented ... and the nth region is cleared only. The clear updates

the depth and/or stencil buffers to the new values and the frame count buffer with the reference value. This region is much smaller than the full window and hence takes less time to clear.

Murphy at col. 25, lines 21-33.

Applicant does not purport herein to have invented striping. While Applicant's invention may be employed in conjunction with striping, striping does not anticipate Applicant's invention as claimed.

2. The Standard for Anticipation

Anticipation requires that each and every element of the claimed invention be disclosed in a single prior art reference. In re Paulsen, 31 U.S.P.Q.2d 1671 (Fed. Cir. 1994); In re Spada, 15 U.S.P.Q.2d 1913 (Fed. Cir. 1990); In re Donohue, 226 U.S.P.Q. 619 (Fed. Cir. 1985). The corollary of this rule is that absence from the reference of any claimed element negates anticipation. Kloster Speedsteel AB v. Crucible Inc., 230 U.S.P.Q. 81 (Fed. Cir. 1986).

Not only must each of the claim elements be disclosed in the single prior art reference, they must also be arranged in the reference as they are arranged in the claim. Richardson v. Suzuki Motor Co., 9 U.S.P.Q.2d 1913 (Fed. Cir. 1989); Carella v. Starlight Archery & Pro Line Co., 231 U.S.P.Q. 644 (Fed. Cir. 1986); Lindemann Maschinenfabrik v. American Hoist & Derrick Co., 221 U.S.P.Q. 481 (Fed. Cir. 1984); Connell v. Sears, Roebuck & Co., 220 U.S.P.Q. 193 (Fed. Cir. 1983). In short, anticipation requires identity between what is claimed and what is disclosed in the reference. Tyler Refrigeration v. Kysor Indus. Corp., 227 U.S.P.Q. 845 (Fed. Cir. 1987); Shatterproof Glass Corp. v. Libbey-Owens Ford Co., 225 U.S.P.Q. 634 (Fed. Cir. 1985); Rosemount, Inc. v. Beckman Instruments, Inc., 221 U.S.P.Q.

1 (Fed. Cir. 1984); Kalman v. Kimberly-Clark Corp., 218 U.S.P.Q. 781 (Fed. Cir. 1983).

3. Claims 1, 4-9 and 47

A. The term "subregion" as used in Murphy refers to one of the n stripes used in the striping fast clear technique.

Appellant's claims 1, 4-9 and 47 presuppose a region having a subregion. As the examiner points out, Murphy uses the term "subregion" at col. 60, line 3. But the term "subregion" as used in Murphy refers to one of the n stripes used in the striping fast clear technique described herein above:

> [T]he interface unit ... performs an effective block clear by incrementing the reference frame counter to indicate a new framecount value, and for addresses in the subregion corresponding to the new framecount value, setting the framecount data equal to the new framecount value, and clearing at least some other portions of the respective data values ... .

Murphy at col. 59, line 67 to col. 60, line 7. In light of the remainder of the Murphy specification, no other interpretation can be fairly supported. See, e.g., Murphy's detailed description of his fast clear functionality at column 55:

> The area that the application is rendering to ... is divided up into n regions, where n is the range of the frame counter ... . Every time the application issues a clear command the reference FrameCount is incremented ... and only the ith region is cleared.
>
> ... [A]t the same time the FrameCount buffer for every pixel in the ith region

is updated with the latest reference FrameCount value. *The region is smaller than the full region the application specifies to be cleared,* so only S/n pixels need to be written. This is roughly n times faster than clearing the full S pixels.

Murphy at col. 55, lines 10-24 (emphasis added). See also Murphy's detailed description of his fast clear functionality at col. 25, lines 29-33, quoted herein above: "[T]he nth region is cleared only. ... *This region is much smaller than the full window and hence takes less time to clear.*" (Emphasis added.)

B. Murphy does not disclose eliminating stale information from pixels outside the nth stripe in response to a first clear command. Nor does Murphy disclose writing an updated clear count into the clear count values of pixels outside the nth stripe in response to a subsequent clear command.

Appellant's claims 1, 4-9 and 47 introduce the notions of a *first* clear command and *subsequent* clear commands directed to the region. The claims require (1) *eliminating stale information from all pixels outside the subregion*[1] responsive to a first clear command, and (2) responsive to a subsequent clear command, updating a current clear count for the region and *writing the updated current clear count into clear count values associated with all pixels outside the subregion.*

Assume for the sake of argument that the nth stripe of Murphy (the stripe corresponding to the value of Murphy's frame counter) is considered to be the "subregion" and that the collection of n stripes is considered to be the "region." Under this assumption,

[1] Appellant has defined in its specification what it means to eliminate stale information from a region. It means writing a predetermined value into only those pixels whose clear counts are not current, but leaving intact the values of pixels whose clear counts are current. See application at page 8, line 1 to page 9, line 10.

the pixels in the other n - 1 stripes would constitute "pixels outside the subregion," and the identity of the nth stripe will necessarily change each time the frame counter in Murphy is updated.

Next assume that application software issues a first clear command to the region and then issues subsequent clear commands to the region. According to the technique of Murphy, the following would occur in response to the first clear command: A clear count for the region (the frame counter) would be updated; all pixels in the subregion would be written with a clear value; all pixel clear count values in the subregion would be set equal to the updated clear count value for the region; and the *pixels and clear count values outside the subregion would not be touched.* See Murphy at col. 25, lines 26-33. Thus, Murphy does not disclose eliminating stale information from all pixels outside the subregion responsive to a first clear command, as is required by Appellant's claims.

Also according to the technique of Murphy, the following would occur in response to a subsequent clear command: The clear count for the region would be updated again, changing the identity of the nth stripe and thus the identity of the subregion; all pixels in the new subregion would be written with a clear value; all pixel clear count values in the new subregion would be set equal to the updated clear count value for the region; and the pixels and clear count values outside the subregion would not be touched. Thus, Murphy does not disclose writing the updated current clear count into clear count values associated with all pixels outside the original subregion responsive to a subsequent clear command, as is required by Appellant's claims.

We cannot logically make the reverse assumption that the n - 1 stripes (instead of the nth stripe) constitute the "subregion," because then the subregion would not be "much smaller than the full window" as is stressed in the Murphy specification. See, e.g., Murphy at col. 25, lines 32-33. Even if we did make this reverse assumption, however, Murphy would still fail to anticipate Appellant's claim because (1) the nth stripe would constitute

"pixels outside the subregion," and (2) Murphy does not disclose eliminating stale information from pixels in the nth stripe. To the contrary–in Murphy, under our reverse assumption, the pixels outside the subregion would all be written with a predetermined clear value. Thus, the values of pixels whose clear counts are current would not be preserved.

Consequently, Murphy does not anticipate Appellant's claims 1, 4-9 or 47 under either of the two interpretations. The rejection should therefore be withdrawn with respect to claims 1, 4-9 and 47.

4. Claims 15, 20-25 and 48

Claims 15, 20-25 and 48 are not anticipated for all of the reasons stated above in relation to claims 1, 4-9 and 47. In addition, claims 15, 20-25 and 48 introduce the additional limitation of: *prior to creation of the subregion,* responding to clear commands according to a fast clear technique wherein a current clear count for the region is updated responsive to each clear command. Nowhere does Murphy disclose operating according to a first fast clear technique prior to creating a subregion and then operating according to a second fast clear technique after creating the subregion. <u>Moreover, the examiner has not alleged that Murphy discloses this limitation</u>. Consequently, the rejection should be withdrawn with respect to claims 15, 20-25 and 48.

5. Claims 31, 36-41 and 50

Claims 31, 36-41 and 50 are not anticipated for all of the reasons stated above in relation to claims 1, 4-9 and 47. In addition, claims 31, 36-41 and 50 introduce the additional limitation of: *determining the percentage area of the region occupied by the subregion; and if the percentage is higher than a predetermined threshold percentage,* responding to clear

commands according to the techniques described herein above. The examiner has not directly addressed this limitation in claim 31. Instead, the examiner alleged that claims 32 and 33 are anticipated by a discussion in Murphy about alpha blending at col. 48, lines 11-15. (Claims 32 and 33 specify particular values for the predetermined threshold of claim 31.)

Alpha blending is a well-known technique in which a new pixel value is "blended" with an existing pixel value in greater or lesser degrees according to an alpha value. The technique is commonly used to produce visual transparency effects in rendered scenes. The technique of alpha blending has nothing to do with the technology of fast clear, and the cited excerpt from Murphy does not purport to suggest any such connection. Rather, the excerpt appears to have been cited simply because it contains the word "percentage." This portion of Murphy is completely inapposite to the examination of Appellant's claims and therefore fails to support the rejection that was asserted against claims 31, 36-41 and 50.

Consequently, the rejection of these claims should be withdrawn.

6. Other Claims

Claims 2, 3, 18, 19, 34, 35 add further detail to an "initialization routine" recited in claims 1, 15 and 31. The initialization routine is to be performed in response to a first clear command. Claims 2, 18 and 34 recite that the initialization routine includes *ensuring that clear count values associated with all pixels inside and outside the subregion are the same.* Claims 3, 19 and 35 recite that the initialization routine includes, after updating the current clear count for the region, *writing the updated current clear count into the clear count values associated with all pixels outside the subregion.* Appellant established above that, in the technique of Murphy, clear count values for pixels outside the subregion are never touched. Therefore, Murphy fails to anticipate any of claims 2, 3, 18, 19, 34 and 35. Consequently, the rejection of these claims should be withdrawn.

Claims 10, 26 and 42 require using an area fill operation to write the updated clear count into the clear count values associated with all pixels outside the subregion. Appellant established above that, in the technique of Murphy, clear count values for pixels outside the subregion are never touched. Therefore, Murphy certainly does not disclose using an area fill operation to touch them. The rejections of claims 10, 26 and 42 should therefore be withdrawn.

Claims 11, 27 and 43 require using a block transfer operation to eliminate stale information from all pixels outside the subregion. Appellant established above that Murphy does not disclose eliminating stale information from pixels outside the subregion according to any technique. Therefore, Murphy certainly does not disclose using a block transfer operation to do so. Consequently, the rejection of claims 11, 27 and 43 should be withdrawn.

Claims 12-14, 28-30 and 44-46 all require employing a first striping technique in the subregion. Appellant has established above that Murphy's "subregion" actually corresponds to one of the n stripes in a striping technique. Murphy does not disclose using any striping technique *within the subregion*. (In the context of Murphy, this would mean using a striping technique *within one of the n stripes*.) Therefore, Murphy does not anticipate any of claims 12-14, 28-30 or 44-46. Consequently, the rejection of those claims should be withdrawn.

Claims 16 and 49 require discontinuing the subregion and resuming responding to clear commands according to the first fast clear technique. Appellant established above that Murphy does not disclose operating according to a first fast clear technique prior to creating a subregion and then operating according to a second fast clear technique after creating the subregion. Therefore, Murphy certainly does not disclose resumption of a first technique after discontinuing a subregion. Consequently, the rejection of claims 16 and 49 should be

withdrawn.

Claim 17 requires that the first fast clear technique be a striping technique, and that the stripe definitions are not changed following resumption of the first technique. Appellant established above that Murphy does not disclose operating according to a first fast clear technique prior to creating a subregion and then operating according to a second fast clear technique after creating the subregion. Therefore, Murphy certainly does not disclose that the first technique should be a striping technique, or that stripe definitions should not change following resumption of the first technique after discontinuing the subregion. Consequently, the rejection of claim 17 should be withdrawn.

Claims 32 and 33 require that the threshold percentage be about 75% and about 70%, respectively. Appellant established above that the examiner's basis for rejecting claim 31 was inapposite to the subject matter of the claims. Moreover, specific percentages were neither recited nor suggested in Murphy–not even in the inapposite context of alpha blending. Therefore, Murphy does not anticipate either of claims 32 and 33. Consequently, the rejection of those claims should be withdrawn.

Conclusion

In view of the above, Appellant requests reversal of the examiner's 11/17/2003 decision finally rejecting claims 1-50.

Respectfully submitted,

Kevin M. Hart

Attorney for Appellant

Reg. No. 36,823

(970)898-7057

Date: 5/17/2004

## APPENDIX

### Claims on Appeal

1. A method of performing clear operations in a region having a subregion, comprising:
performing an initialization routine responsive to a first clear command, wherein the initialization routine comprises eliminating stale information from all pixels outside the subregion; and
responsive to a subsequent clear command:
updating a current clear count for the region; and
writing the updated current clear count into clear count values associated with all pixels outside the subregion.

2. The method of claim 1, wherein the initialization routing further comprises:
ensuring that clear count values associated with all pixels inside and outside the subregion are the same; and
writing a predetermined value into all pixels inside the subregion.

3. The method of claim 1, wherein the initialization routine further comprises:
updating the current clear count for the region; and
writing the updated current clear count into the clear count values associated with all pixels outside the subregion.

4. The method of claim 1, wherein the pixels correspond to an image buffer.

5. The method of claim 1, wherein the pixels correspond to a z buffer.

6. The method of claim 1, wherein:

the method is performed using a fast clear computer graphics system in which a predetermined color value is displayed in lieu of a pixel color value for pixels whose associated clear count value does not equal the current clear count; and

the predetermined color value is the same as a background color outside the subregion.

7. The method of claim 1, wherein:

the method is performed using a fast clear computer graphics system in which a predetermined color value is displayed in lieu of a pixel color value for pixels whose associated clear count value does not equal the current clear count; and

the predetermined color value is not the same as a background color outside the subregion.

8. The method of claim 1, wherein the subregion is a scissor region.

9. The method of claim 1, wherein the subregion is a viewport.

10. The method of claim 1, further comprising:

dividing an area outside the subregion into at least one rectangular subarea; and

wherein writing the updated current clear count into the clear count values associated with all pixels outside the subregion comprises executing an area fill operation on the at least one rectangular subarea.

11. The method of claim 1, further comprising:

dividing an area outside the subregion into at least one rectangular subarea; and

wherein eliminating stale information from all pixels outside the subregion comprises executing a block transfer operation on the at least one rectangular subarea, wherein

a source region and a destination region for the block transfer operation both correspond to the at least one rectangular subarea.

12. The method of claim 1, further comprising:
employing a first striping technique in the subregion.

13. The method of claim 12, further comprising:
employing a second striping technique in the region prior to creation of the subregion;
wherein the first and second striping techniques are the same striping techniques; and
wherein stripe definitions used for striping in the subregion are the same as those that were created and used for striping in the region prior to creation of the subregion.

14. The method of claim 12, further comprising:
employing a second striping technique in the region prior to creation of the subregion;
wherein the first and second striping techniques are the same striping techniques; and
wherein stripe definitions used for striping in the subregion are not the same as those that were created and used for striping in the region prior to creation of the subregion.

15. A method of performing clear operations in a region having a subregion, comprising:

prior to creation of the subregion, responding to clear commands according to a fast clear technique wherein a current clear count for the region is updated responsive to each clear command;

responsive to a first clear command after creation of the subregion:

performing an initialization routine comprising eliminating stale information from all pixels outside the subregion; and

responsive to a subsequent clear command after creation of the subregion:

updating the current clear count for the region; and

writing the updated current clear count into the clear count values associated with all pixels outside the subregion.

16. The method of claim 15, further comprising:

discontinuing the subregion; and

resuming responding to clear commands according to the fast clear technique.

17. The method of claim 16, wherein:

the fast clear technique used prior to creation of the subregion is a striping technique; and wherein stripe definitions for the striping technique are not changed in the resuming step.

18. The method of claim 15, wherein the initialization routing further comprises:

ensuring that clear count values associated with all pixels inside and outside the subregion are the same; and

writing a predetermined value into all pixels inside the subregion.

19. The method of claim 15, wherein the initialization routine further comprises:

updating the current clear count for the region; and

writing the updated current clear count into the clear count values associated with all pixels outside the subregion.

20. The method of claim 15, wherein the pixels correspond to an image buffer.

21. The method of claim 15, wherein the pixels correspond to a z buffer.

22. The method of claim 15, wherein:

the method is performed using a fast clear computer graphics system in which a predetermined color value is displayed in lieu of a pixel color value for pixels whose associated clear count value does not equal the current clear count; and

the predetermined color value is the same as a background color outside the subregion.

23. The method of claim 15, wherein:

the method is performed using a fast clear computer graphics system in which a predetermined color value is displayed in lieu of a pixel color value for pixels whose associated clear count value does not equal the current clear count; and

the predetermined color value is not the same as a background color outside the subregion.

24. The method of claim 15, wherein the subregion is a scissor region.

25. The method of claim 15, wherein the subregion is a viewport.

26. The method of claim 15, further comprising:

dividing an area outside the subregion into at least one rectangular subarea; and

wherein writing the updated current clear count into the clear count values associated with all pixels outside the subregion comprises executing an area fill operation on the at least one rectangular subarea.

27. The method of claim 15, further comprising:
dividing an area outside the subregion into at least one rectangular subarea; and
wherein eliminating stale information from all pixels outside the subregion comprises executing a block transfer operation on the at least one rectangular subarea, wherein a source region and a destination region for the block transfer operation both correspond to the at least one rectangular subarea.

28. The method of claim 15, further comprising:
employing a first striping technique in the subregion.

29. The method of claim 28, further comprising:
employing a second striping technique in the region prior to creation of the subregion;
wherein the first and second striping techniques are the same striping techniques; and
wherein stripe definitions used for striping in the subregion are the same as those that were created and used for striping in the region prior to creation of the subregion.

30. The method of claim 28, further comprising:
employing a second striping technique in the region prior to creation of the subregion;
wherein the first and second striping techniques are the same striping techniques; and
wherein stripe definitions used for striping in the subregion are not the same as those that were created and used for striping in the region prior to creation of the subregion.

31. A method of performing clear operations in a region having a subregion, comprising:
determining the percentage area of the region occupied by the subregion; and
if the percentage area is higher than a predetermined threshold percentage, responding to clear commands by:
performing an initialization routine responsive to a first clear command, wherein the initialization routine comprises eliminating stale information from all pixels outside the subregion; and
responsive to a subsequent clear command:
updating a current clear count for the region; and
writing the updated current clear count into clear count values associated with all pixels outside the subregion.

32. The method of claim 31, wherein the predetermined threshold percentage is about 75%.

33. The method of claim 31, wherein the predetermined threshold percentage is about 70%.

34. The method of claim 31, wherein the initialization routing further comprises:
ensuring that clear count values associated with all pixels inside and outside the subregion are the same; and
writing a predetermined value into all pixels inside the subregion.

35. The method of claim 31, wherein the initialization routine further comprises:
updating the current clear count for the region; and
writing the updated current clear count into the clear count values associated with all pixels

outside the subregion.

36. The method of claim 31, wherein the pixels correspond to an image buffer.

37. The method of claim 31, wherein the pixels correspond to a z buffer.

38. The method of claim 31, wherein:

the method is performed using a fast clear computer graphics system in which a predetermined color value is displayed in lieu of a pixel color value for pixels whose associated clear count value does not equal the current clear count; and

the predetermined color value is the same as a background color outside the subregion.

39. The method of claim 31, wherein:

the method is performed using a fast clear computer graphics system in which a predetermined color value is displayed in lieu of a pixel color value for pixels whose associated clear count value does not equal the current clear count; and

the predetermined color value is not the same as a background color outside the subregion.

40. The method of claim 31, wherein the subregion is a scissor region.

41. The method of claim 31, wherein the subregion is a viewport.

42. The method of claim 31, further comprising:

dividing an area outside the subregion into at least one rectangular subarea; and

wherein writing the updated current clear count into the clear count values associated with all pixels outside the subregion comprises executing an area fill operation on the at

least one rectangular subarea.

43. The method of claim 31, further comprising:
dividing an area outside the subregion into at least one rectangular subarea; and
wherein eliminating stale information from all pixels outside the subregion comprises executing a block transfer operation on the at least one rectangular subarea, wherein a source region and a destination region for the block transfer operation both correspond to the at least one rectangular subarea.

44. The method of claim 31, further comprising:
employing a first striping technique in the subregion.

45. The method of claim 31, further comprising:
employing a second striping technique in the region prior to creation of the subregion;
wherein the first and second striping techniques are the same striping techniques; and
wherein stripe definitions used for striping in the subregion are the same as those that were created and used for striping in the region prior to creation of the subregion.

46. The method of claim 31, further comprising:
employing a second striping technique in the region prior to creation of the subregion;
wherein the first and second striping techniques are the same striping techniques; and
wherein stripe definitions used for striping in the subregion are not the same as those that were created and used for striping in the region prior to creation of the subregion.

47. Computer program code embodied in a machine-readable storage or transmission medium that, when executed on a computer, causes the computer to perform a method of performing clear operations in a region having a subregion, comprising:

performing an initialization routine responsive to a first clear command, wherein the initialization routine comprises eliminating stale information from all pixels outside the subregion; and

responsive to a subsequent clear command:

updating a current clear count for the region; and

writing the updated current clear count into clear count values associated with all pixels outside the subregion.

48. Computer program code embodied in a machine-readable storage or transmission medium that, when executed on a computer, causes the computer to perform a method of performing clear operations in a region having a subregion, comprising:

prior to creation of the subregion, responding to clear commands according to a fast clear technique wherein a current clear count for the region is updated responsive to each clear command;

responsive to a first clear command after creation of the subregion:

performing an initialization routine comprising eliminating stale information from all pixels outside the subregion; and

responsive to a subsequent clear command after creation of the subregion:

updating the current clear count for the region; and

writing the updated current clear count into the clear count values associated with all pixels outside the subregion.

49. The computer program code of claim 48, wherein the method further comprises:

discontinuing the subregion; and

resuming responding to clear commands according to the fast clear technique.

50. Computer program code embodied in a machine-readable storage or transmission medium that, when executed on a computer, causes the computer to perform a method of performing clear operations in a region having a subregion, comprising:

determining the percentage area of the region occupied by the subregion; and

if the percentage area is higher than a predetermined threshold percentage, responding to clear commands by:

performing an initialization routine responsive to a first clear command, wherein the initialization routine comprises eliminating stale information from all pixels outside the subregion; and

responsive to a subsequent clear command:

updating a current clear count for the region; and

writing the updated current clear count into clear count values associated with all pixels outside the subregion.